REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE 1/31/94

3. REPORT TYPE AND DATES COVERED Technical Report 7/1/91-11/30/94

4. TITLE AND SUBTITLE
Dynamical Systems, Neural Networks and Cortical Models

5. FUNDING NUMBERS
~~F~~AFOSR-91-0325

6. AUTHOR(S)
Morris W. Hirsch and William Baird

AFOSR-TR 95-0389

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center For Pure and Applied Mathematics
University of California
Berkeley, CA 94720

8. PERFORMING ORGANIZATION REPORT NUMBER
1-443964-22506

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
110 Duncan Avenue Suite B115
Bolling Air Force Base
Washington, D.C. 20332-0001

[illegible]

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
~~F~~AFOSR-91-0325

11. SUPPLEMENTARY NOTES

This document has been approved for public release and sale; its distribution is unlimited.

DTIC
SELECTE
JUN 1 6 1995
S D
F

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT

This work developed new biological models of cortex and produced artificial neural systems for computation which exploited the special capabilities of complex dynamics, and applied them to specific engineering problems: handwritten character recognition, and grammatical inference.

Associative memory modules with oscillatory and chaotic attractors were successfully applied to the problem of handwritten character recognition in early stages of the work.

In the later stages of the project we succeeded in demonstrating analytically and numerically how an unusual cortical "sensory-motor" computing architecture, can be constructed of recurrently interconnected modules of this type. These learn connection weights between themselves which cause the system to evolve under a clocked "machine cycle" by a sequence of transitions of attractors within the modules, much as a digital computer evolves by transitions of its binary flip-flop states.

Because intercommunicating modules of the architecture are analytically guaranteed to store and recall multiple oscillatory and chaotic attractors, the architecture has served as a framework in which to arrange and exploit the special capabilities dynamic attractors.

Most recently we showed how the architecture can learn to employ selective "attentional" control of synchronization to direct the flow of communication and computation within the architecture to solve a grammatical inference problem.

This type of computing architecture and its learning algorithms for computation with oscillatory spatial modes is ideal for implementation in optical systems, where electromagnetic oscillations, very high dimensional modes, and high processing speeds are available.

14. SUBJECT TERMS

DTIC QUALITY INSPECTED 3

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

19950614 050

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	Contract	PR	Project
G	Grant	TA	Task
PE	Program Element	WU	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	See authorities.
NASA	See Handbook NHB 2200.2.
NTIS	Leave blank.

Block 12b. Distribution Code.

DOD	Leave blank.
DOE	Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA	Leave blank.
NTIS	Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

FINAL TECHNICAL REPORT

Dynamical Systems, Neural Networks, and Cortical Models

Morris W. Hirsch (PI)
Dept Mathematics, U.C.Berkeley,
Berkeley, Ca. 94720, 510-642-4318

Bill Baird (Assist. Research Mathematician)
Dept Mathematics, U.C. Berkeley

Walter Freeman (Collaborator)
Dept Molecular and Cell Biology, U.C. Berkeley

Mathematics resource manager: Bernice Gangale 510-642-0116

Period: 7/1/91 -11/30/94

Principal Investigator - Morris Hirsch

Department chair - John B. Wagoner

Director of the Center for Pure and Applied Math - Alan Weinstein

Contact and Grant officer - Pat Gates

Abstract

This work developed new biological models of cortex and produced artificial neural systems for computation which exploited the special capabilities of complex dynamics, and applied them to specific engineering problems: handwritten character and word recognition, and grammatical inference.

In the later stages of the project we succeeded in demonstrating analytically and numerically how an unusual cortical "sensory-motor" computing architecture, can be constructed of recurrently interconnected associative memory modules. Because intercommunicating modules of the architecture are analytically guaranteed to store and recall multiple oscillatory and chaotic attractors, the architecture has served as a framework in which to arrange and exploit the special capabilities dynamic attractors. Modules with oscillatory and chaotic attractors were successfully applied to the problem of handwritten character recognition in early stages of the work.

The modules in the larger architecture can learn connection weights between themselves which cause the system to evolve under a clocked "machine cycle" by a sequence of transitions of attractors within the modules, much as a digital computer evolves by transitions of its binary flip-flop states. The architecture thus employs the principle of "computing with attractors" used by macroscopic systems for reliable computation in the presence of noise.

Superior noise immunity was demonstrated for these systems with dynamic attractors over systems with static attractors, and synchronization between coupled periodic or chaotic attractors in different modules was shown to be important for effecting reliable transitions. We synchronized chaotic attractors for operation in the architecture using techniques of coupling developed for secure "broadspectrum" communication by a modulated chaotic carrier wave.

We constructed a system which learns to function as a finite state automaton that perfectly recognizes or generates the infinite set of six symbol strings that are defined by a Reber grammar. Even though it is constructed from a system of continuous nonlinear ordinary differential equations, the system can operate as a discrete-time symbol processing architecture, but with analog input and oscillatory subsymbolic representations.

Most recently we showed how the architecture can learn to employ selective "attentional" control of synchronization to direct the flow of communication and computation within the architecture to solve a more difficult grammatical inference problem.

This type of computing architecture and its learning algorithms for computation with oscillatory spatial modes is ideal for implementation in optical systems, where electromagnetic oscillations, very high dimensional modes, and high processing speeds are available.

Contents

1 Summary of Work Done

Over the period of this grant, AFOSR-91-0325, entitled "Dynamical Systems, Neural Networks, and Cortical Models", we published 10 papers and one book chapter [5, 7, 8, 10, 9, 12, 11, 14, 13, 29, 15]. Fifteen conferences were attended where oral or poster presentations were made - including nine invited talks around the world in France, Japan, and Austrailia, as detailed in a later section.

The mathematical foundation for the work is a learning algorithm for recurrent analog neural networks, the **normal form projection algorithm**, developed at Berkeley on this grant. It allows analytically guaranteed associative memory storage of analog patterns, continuous sequences, and chaotic attractors in the same network. We know of no other system with such a guarantee. There are N units of capacity in an N unit network. It costs one unit per static attractor, two per Fourier component of each periodic sequence (oscillating attractor), and three (or more) per chaotic attractor. There are no spurious attractors. For periodic sequences there is a Liapunov function which governs the approach of transient states to stored trajectories [12].

We showed that the general projection learning rule reduced to a Hebbian outer product rule for storage of orthogonal patterns in a biological model of olfactory cortex. We showed further that a model of coupled oscillatory neural populations which assumed only minimal coupling justified by known anatomy resulted if the oscillating patterns were further restricted to have the phase structure of those observed experimentally in olfactory cortex [12].

An alternative network for for implementation of the projection algorithm called the "projection network" [9] was then developed. This network has $3N^2$ weights instead of $N^2 + N^4$, and is more useful for engineering applications and for simulations of the biological model.

These networks, including some with chaotic attractors, were successfully applied to the problem of real time handwritten character recognition. This is still the first system we know of that can accomplish reliable pattern recognition with exclusively chaotic dynamics [7].

The biological foundations of the network were deepened when we showed how the higher order weights of the biological model could be realized by synaptic clusters on dendrites in the neuropil [5]. We showed as well how the Hebbian multiple outer product rule could be decomposed to reveal a term called "competition" that could control attractor transitions. It could be realized biologically by a nucleus that summed all network activations and fedback a global inhibition to all nodes of the network [5].

In the next project, we showed analytically and numerically how a cortical "sensory-motor" computing architecture, could be constructed of recurrently interconnected associative memory modules [10]. The architecture is such that the larger system is itself a special case of the type of network of the modules, and can be analysed with the same tools used to design the subnetwork modules. Because intercommunicating modules of the architecture can store and recall multiple oscillatory and chaotic attractors, the architecture serves as a framework in which to arrange and exploit the special capabilities dynamic attractors [?].

The modules in the larger architecture learn connection weights which cause the system to evolve under a clocked "sensory-motor cycle" by a sequence of transitions of attractors within the modules, much as a digital computer evolves by transitions of its binary flip-flop attractors. This architecture employs the principle of "computing with attractors" used by macroscopic systems for reliable computation in the presence of noise. The competition parameter mentioned above is used as a bifurcation parameter to clock the attractor transitions of the sensory-motor cycle [10].

We then constructed a discrete-time "simple recurrent" or "Elman" network architecture with oscillatory modules. The time steps (machine cycles) of the system hold input and "context" modules clamped at their oscillatory attractors while "hidden" modules change state, then clamp hidden states while context modules are released to load those states as the new context for the next cycle of input [14].

The system learned to function as a finite state automaton that perfectly recognizes, or alternatively generates, the infinite set of six symbol strings defined by a Reber grammar. Even though it is constructed from a system of continuous nonlinear ordinary differential equations, the system operates as a discrete-time symbol processing architecture, but with analog input and oscillatory subsymbolic representations.

Superior noise immunity was demonstrated for modules with dynamic attractors over modules with static attractors, and synchronization between coupled periodic or chaotic attractors in different modules has been shown to be important for effecting reliable transitions. We synchronized Lorenz attractors for operation in the architecture using "control of chaos" techniques of coupling developed for secure "broadspectrum" communication by a modulated chaotic carrier wave [13].

We demonstrated in recent analysis and simulations that sets of canonical equations for the Chua circuit can be used in the projection network [13]. We have found attractors in the Chua family which out-perform the Lorenz attractor in our handwritten character recognition systems, with fast competitive suppression at low values of competitive coupling.

Most recently we have shown that the Elman architecture can learn to employ selective "attentional" control of synchronization to direct the flow of communication and computation within the architecture to solve a grammatical inference problem [15].

In this architecture, oscillation amplitude codes the information content or activity of a module (unit), whereas phase and frequency are used to "softwire" the network. We have shown that only synchronized modules communicate by exchanging amplitude information; the activity of non-resonating modules is shown to contribute noise. The same hardware and connection matrix can thus subserve many different computations and patterns of interaction between modules.

Synchronization control is modeled as a subset of the hidden modules with ouputs which affect the resonant frequencies of other hidden modules. They learn to perturb these frequencies to control synchrony among these modules and direct the flow of computation to effect transitions between subsections of a large automaton which the system learns to emulate. The internal crosstalk noise is used to drive the required random transitions of the automaton.

2 Introduction

The goal of this work is to produce new systems for computation that exploit the special capabilities of complex dynamics and apply them to specific engineering problems like handwritten character and word recognition, speech recognition, grammatical inference, adaptive system identification and control, autonomous robot navigation, and artificial intelligence.

Recordings of local field potentials have revealed 40 to 80 Hz oscillation in vertebrate cortex [24, 25]. The amplitude patterns of such oscillations have been shown to predict the olfactory and visual pattern recognition responses of a trained animal. There is further evidence that although the oscillatory activity appears to be roughly periodic, it is actually chaotic when examined in detail. This preliminary evidence suggests that oscillatory or chaotic network modules may form the cortical substrate for many of the sensory, motor, and cognitive functions now studied in static networks.

It remains be shown how networks with more complex dynamics can performs these operations and what possible advantages are to be gained by such complexity. Our work has therefore culminated in the construction of a parallel distributed processing architecture that is inspired by the structure and dynamics of cerebral cortex, and applied it to the problem of grammatical inference. The construction views cortex as a set of coupled oscillatory associative memories, and is guided by the principle that attractors must be used by macroscopic systems for reliable computation in the presence of noise.

This system must function reliably in the midst of noise generated by crosstalk from it's own activity. Present day digital computers are built of flip-flops which, at the level of their transistors, are continuous dissipative dynamical systems with different attractors underlying the symbols we call "0" and "1". In a similar manner, the network we have constructed is a symbol processing system, but with analog input and oscillatory subsymbolic representations.

Periodic or nearly periodic (chaotic) variation of a signal introduces a additional degrees of freedom that can be exploited in a computational architecture. We are presently investigating the design principle that selective control of synchronization, which we consider to be a model of "attention", can be used to control program flow in an architecture with dynamic attractors.

The architecture operates as a thirteen state finite automaton that generates the symbol strings of a Reber grammar. It is designed to demonstrate and study the following issues and principles of neural computation: (1) Sequential computation with coupled associative memories. (2) Computation with attractors for reliable operation in the presence of noise. (3) Discrete time and state symbol processing arising from continuum dynamics by bifurcations of attractors. (4) Attention as selective synchronization controling communication and temporal program flow. (5) chaotic dynamics in some network modules driving randomn choice of attractors in other network modules.

We believe that this type of computing architecture and its learning algorithms for computation with oscillatory spatial modes is ideal for implementation in optical systems, where electromagnetic oscillations, very high dimensional modes, and high processing speeds are available. The mathematical expressions for optical mode competition are identical to our normal form equations for oscillatory mode competition.

To advance intuition for theoretical analysis, interactive simulations of the network applications have been designed on the SGI 4D35G Personal Iris Graphics Workstation. These allow real time graphic display of network dynamics and learning as parameters are varied.

3 Mathematical Background

The mathematical foundation for the current project is a learning algorithm for recurrent analog neural networks, the **normal form projection algorithm**, developed at Berkeley on this grant, AFOSR-91-0325. It allows analytically guaranteed associative memory storage of analog patterns, continuous sequences, and chaotic attractors in the same network.

3.1 The Projection Algorithm

A key feature of a net constructed by this algorithm is that the underlying dynamics is explicitly isomorphic to any of a class of standard, well understood nonlinear dynamical systems - a "normal form" [27]. This control over the dynamics permits the design of important aspects of the network dynamics independent of the particular patterns to be stored. Stability, basin geometry, and rates of convergence to attractors can be programmed in the standard dynamical system.

There are N units of capacity in an N unit network. It costs one unit per static attractor, two per Fourier component of each periodic sequence (oscillating attractor), and three (or more) per chaotic attractor. There are no spurious attractors. For periodic sequences there is a Liapunov function which governs the approach of transient states to stored trajectories [4].

We make particular use of the normal form for the **Hopf bifurcation** [27] configured as a simple recurrent competitive k-winner-take-all network with a cubic nonlinearity, shown here in Cartesian coordinate form.

$$\dot{v}_i = \sum_{j=1}^{N} J_{ij} v_j - v_i \sum_{j=1}^{N} A_{ij} v_j^2 \tag{1}$$

This *model dynamical system* is expressed in diagonalized "overlap" or "memory coordinates" (one memory per k nodes). Matrix J is at the disposal of the experimenter: A diagonal matrix with real eigenvalues determines static attractors; alternatively, periodic attractors are obtained if J is the real form of a complex diagonal matrix with positive real parts α. This causes initial states to move away from the origin, until the competitive (negative) cubic terms dominate at some distance, causing the flow to be inward from all points beyond. The off-diagonal cubic terms create competition between directions of flow within a spherical middle region and thus create multiple attractors and basins. The larger the eigenvalues in J and off-diagonal weights in A, the faster the convergence to attractors in this region. For temporal patterns, these nodes come in complex conjugate pairs which supply Fourier components for trajectories to be learned. Many types of dynamics have been implemented by specializing A and J, including static attractors, limit cycles, and chaotic attractors. Chaotic dynamics may be created by specific programming of the interaction of two pairs of these nodes.

The rule for learning desired distributed spatial or spatio-temporal patterns can be shown to be equivalent to the operation of "projecting" sets of these nodes into "network" coordinates (the standard basis) using the desired vectors as corresponding columns of a transformation matrix P. The differential equations of the recurrent network itself may be viewed as linearly transformed or "projected", leading to new recurrent network equations with general coupling T_{ij} for the linear terms, and general higher order weights T_{ijkl} for the cubic terms [1]. The projection learning rule is,

$$T = PJP^{-1}\ ,\ \ and\ \ T_{ijkl} = \sum_{mn=1}^{n} P_{im}\tilde{A}_{mn}P_{mj}^{-1}P_{nk}^{-1}P_{nl}^{-1}. \tag{2}$$

and, in the general case, the normal form (1) is projected to become,

$$\dot{x}_i = -\tau x_i + \sum_{j=1}^{N} T_{ij}x_j - \sum_{jkl}^{N} T_{ijkl}x_j x_k x_l. \tag{3}$$

3.1.1 Chaotic attractors

We use the well studied family of Chua attractors[30, 20], implemented by the Chua hardware circuit[19], to investigate the effectiveness of complex dynamics in this associative memory. This is a physical system whose mathematical model is capable of duplicating most experimentally observed chaotic and bifurcation phenomena, and which has yielded to a mathematical treatment.

The Canonical Chua equations give access to a rich variety of dynamics to explore in the system. Recent work in the Chua group has revealed that more than 30 chaotic attractors including types similar to those that have been studied in the literature, which we have previously employed, such as the Lorenz and Roessler attractors, can be obtained from the cannonical equations for the Chua circuit by variation of 7 parameters [18].

We have demonstrated in recent analysis and simulations that sets of canonical equations for the Chua circuit in three dimensional subspace blocks can be used in the projection network. Multiple Chua attractors have been created simply by adding off-diagonal normal form competitive terms to couple sets of the three Chua equations,

Case I: $RC_2 > 0$

$$\dot{v}_1 = \alpha(v_1 - v_2 - f(v_1)) - v_1 \sum_{j=1}^{N} A_{1j}v_j^2 \tag{4}$$

$$\dot{v}_2 = v_1 - v_2 + v_3 - v_2 \sum_{j=1}^{N} A_{2j}v_j^2 \tag{5}$$

$$\dot{v}_3 = -\beta v_2 - \gamma v_3 - v_3 \sum_{j=1}^{N} A_{3j}v_j^2 \tag{6}$$

$$\vdots \tag{7}$$

Case II: $RC_2 < 0$

$$\dot{v}_1 = \alpha(-v_1 + v_2 + f(v_1)) - v_1 \sum_{j=1}^{N} A_{1j}v_j^2 \tag{8}$$

$$\dot{v}_2 = -v_1 + v_2 - v_3 - v_2 \sum_{j=1}^{N} A_{2j}v_j^2 \tag{9}$$

$$\dot{v}_3 = \beta v_2 + \gamma v_3 - v_3 \sum_{j=1}^{N} A_{3j}v_j^2 \tag{10}$$

$$\vdots \tag{11}$$

$$f(v_1) = bv_1 + 1/2(a-b)(|v_1+1| - |v_1-1|).$$

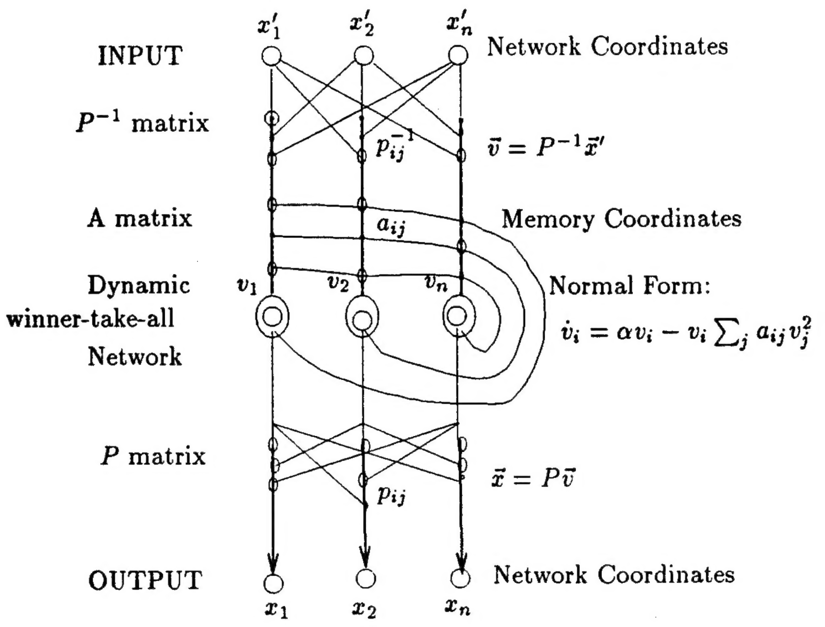

Figure 1: Projection Network with $3N^2$ weights. The A matrix programs a k-winner-take-all net which determines attractors, basins of attraction, and rates of convergence. The columns of P contain the ouptut patterns associated to these attractors. The rows of P^{-1} determine category centroids

The A matrix here has constant off diagonal coupling, and three dimensional blocks of zeros along the diagonal to remove the self-damping terms from the subspaces containing the Chua equations.

If we use the cubic nonlinearity

$$f(v_1) = a_0 v_1 + a_1 v_1^3,$$

then these equations are of the same general form as the Hopf normal form (with a different linear part), and they may be projected into network coordinates by the learning rule (2) exactly as shown above to give network equations (3) The matrix J now contains the linear coupling terms of the Chua equations in three dimensional blocks along the diagonal, and zeros everywhere else. The diagonal blocks J^B in J have the following form,

$$J^B = \begin{bmatrix} (-\alpha + a_0) & \alpha & 0 \\ 1 & -1 & 1 \\ 0 & -\beta & -\gamma \end{bmatrix},$$

Here the matrix A is as described above, except for the the diagonal slot $A_{1i} = \alpha a_1$ from the cubic nonlinear term $\alpha a_1 v_1^3$ of the Chua equations.

3.2 Projection Network

An alternative network for for implementation of the projection algorithm is the "projection network" [7]. In the projection net, the algebraic projection operation into and out of memory coordinates is done explicitly by a set of weights in two feedforward linear networks characterized by weight matrices P^{-1} and P. These map inputs into and out of the nodes of the recurrent dynamical network in memory coordinates sandwiched between them. This kind of network, with explicit input and output projection maps that are inverses, may be considered an "unfolded" version of the purely recurrent networks described above.

The autoassociative case of this network is formally equivalent to the higher order network realization used above as a biological model [3]. All the mathematical results proved for the projection algorithm in this case carry over to this new architecture, but more general versions can be trained and applied in novel ways. The new network has $3N^2$ weights instead of $N^2 + N^4$, and is more useful for engineering applications and for simulations of the biological model. The $2N^2$ input and output weights could be stored off-chip in a conventional memory, and the fixed weights of the dynamic normal form network could be implemented in analog VLSI for fast analog relaxation.

This network is shown in figure 1. Input pattern vectors $\vec{X}'$ are applied as pulses which project onto each vector of weights (row of the P^{-1} matrix) on the input to each unit i of the dynamic network to establish an activation level v_i which determines the initial condition for the relaxation dynamics of this network. The recurrent weight matrix A of the dynamic network can be chosen so that the unit or predefined subspace of units which recieves the largest projection of the input will converge to some state of activity, static or dynamic, while all other units are supressed to zero activity.

Given prototype patterns to be stored, a matrix inversion determines network weights. For nearly orthogonal patterns, matrix transposition is used instead. Unsupervised or supervised incremental learning algorithms for pattern classification, such as competitive learning or bootstrap Widrow-Hoff can easily be implemented. We have well behaved simulations containing multiple static, oscillatory, and chaotic attractors in different competing subspaces of the same network.

3.2.1 Handwritten Character Recognition

Using the projection architecture, an effective real time handwritten character recognition system with mouse input of characters and on line learning has been developed. Various options allow the system to utilize static, oscillatory, and/or chaotic attractors (Lorenz, Rossler, Ruelle-Takens, Chua, etc). This is the first system we know of that can accomplish reliable pattern recognition with exclusively chaotic dynamics [7].

Handwritten characters have a natural scale and translation invariant analog representation in terms of a sequence of angles that parametrize the pencil trajectory. We remove the writing velocity variation from the raw set of (x,y) samples of a digit trajectory by interpolating a new set of N (x,y) points equally spaced along the curve. A vector of the sines and cosines of the angles from the horizontal made by the line segment emanating from each point then becomes the preprocessed representation of the digit to be learned or input to the network for recognition.

Learning in this system can be as fast as recognition. We have seen that the projection algorithm supplies a formula that allows one shot "learning" of prototypes and immediately establishes a basin of attraction that determines the generalization response of the network to future inputs. Using a projection network architecture with 32 attractors for digits and lower case letters, where the input vector is of dimension N = 64, with only this one shot learning of prototypes, and very small databases for a single writer at a time, all attractors allow roughly 95% correct recognition responses. Unexpected properties have been found in the systems utilizing chaotic attractors. Chaotic attractors, for example have different basins of attraction from static or periodic attractors.

In the projection network, or its folded biological version, the chaotic attractors have a basin of attraction in the N dimensional state space that constitues a category, just like any other attractor in the system. There may be computational advantages to the basins of attraction (categories) produced by chaotic attractors, or to the effects their outputs have as inputs to other network modules.

In the projected or folded network coordinates, the particular distrubuted N dimensional spatio-temporal patterns learned for the four components of the chaotically paired oscillatory modes, or the three components of the Lorenz system, form a coordinate-specific "encoding" of the chaotic attractor, which may constitute a recognizable input to another network, if it falls within some learned basin of attraction. While the details of the trajectory of a chaotic attractor in any physical continuous dynamical system are lost in the noise, there is still a particular structure to the attractor which is a recognizable "signature". This allows communication and "recognition" of chaotic attractors.

We have found attractors in the Chua family which out-perform the Lorenz attractor in our handwritten character recognition systems, with fast competitive suppression at low values of competitive coupling.

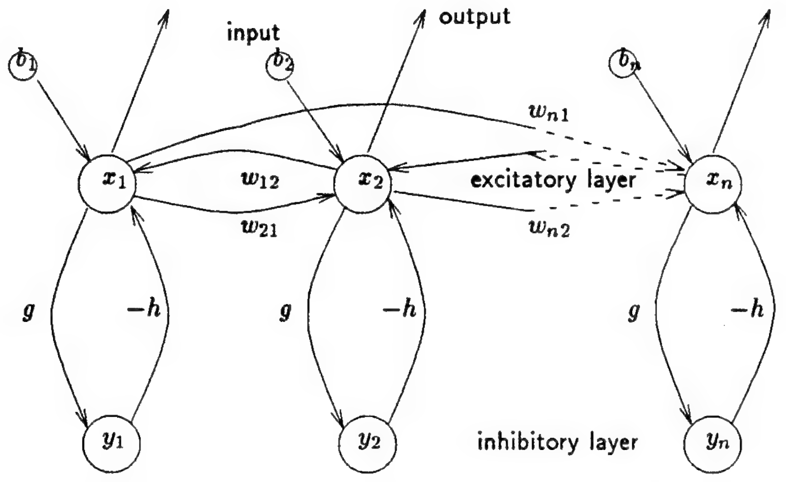

Figure 2: Biological subnetwork of excitatory cell populations x_i, inhibitory cell populations y_i, inputs b_i, adaptive excitatory to excitatory connections W_{ij}, and constant local inhibitory feedback connections g and $-h$.

3.3 Biological Associative Memory Module

We have determined a biologically "minimal" model that is intended to assume the least anatomically justified coupling sufficient to allow function as an oscillatory associative memory. The network shown in figure 2 is meant only as a cartoon of the real biology, which is designed to reveal the general mathematical principles and mechanisms by which the actual system might function.

Long range excitatory to excitatory connections are well known as "associational" connections in olfactory cortex[28] and cortico-cortico connections in neocortex. Since our units are neural populations, we can expect that some density of full cross-coupling exists in the system[28], and our weights are taken to be the average synaptic strengths of these connections. Local inhibitory "interneurons" are a ubiquitous feature of the anatomy of cortex [26, 23]. It is unlikely that they make long range connections (> 1 mm) by themselves. These connections, and even the debated interconnections between them, are therefore left out of a minimal coupling model. The resulting network is a fair cartoon of the well studied circuitry of olfactory (pyriform) cortex. Since almost all of cortex has this type of structure in the brains of amphibia and reptiles, our super-network of these submodules has the potential to become a reasonable caricature of the full cortical architecture in these animals. Although the neocortex of mammals is more complicated, we expect the model to provide useful suggestions about the principles of oscillatory computation there as well.

For an N dimensional system, this minimal coupling structure is described mathematically by the matrix T.

$$T = \begin{bmatrix} W & -hI \\ gI & 0 \end{bmatrix} \tag{12}$$

W is the $N/2 \times N/2$ matrix of excitatory interconnections, and gI and hI are $N/2 \times N/2$ identity matrices multiplied by the positive scalars g, and h. These give the strength of coupling around local inhibitory feedback loops. A state vector is composed of local average cell voltages for $N/2$ excitatory neuron populations $\vec{x}$ and $N/2$ inhibitory neuron populations $\vec{y}$. Intuitively, since the inhibitory units receive no direct input and give no direct output, they act as hidden units that create oscillation for the amplitude patterns stored in the excitatory cross-connections W. This may perhaps be viewed as a specific structural addition to a recurrent analog higher order network architecture to convert its static attractors to periodic attractors. Here the symmetric sigmoid functions of such a network are Taylor expanded up to cubic terms with third order weights (quadratic terms are killed by the symmetry). Network equations with the first order coupling

(12) shown above, plus these higher order excitatory synapses, are shown below, in component form.

$$\dot{x}_i = -\tau x_i - h y_i + \sum_{j=1}^{N/2} W_{ij} x_j - \sum_{jkl=1}^{N/2} W_{ijkl} x_j x_k x_l + b_i, \tag{13}$$

$$\dot{y}_i = -\tau y_i + g x_i, \tag{14}$$

The competitive (negative) cubic terms of constitute a directly programmable nonlinearity that is independent of the linear terms. Normal form theory shows that these cubics are the essential nonlinear terms required to store oscillations, because of the (odd) phase shift symmetry required in the vector field. They serve to create multiple periodic attractors by causing the oscillatory modes of the linear term to compete, much as the sigmoidal nonlinearity does for static modes in a network with static attractors [1, 4]. Intuitively, these terms may be thought of as sculpting the maxima of a "saturation" (energy) landscape, into which the modes with positive eigenvalues expand, and positioning them to lie in the directions specified by the eigenvectors to make them stable. A Liapunov function for this landscape may be explicitly constructed in a special polar coordinate system [1, 4]. We use this network directly as our biological model. From a physiological point of view, (13) may be considered a model of a biological network which is operating in the linear region of the known *axonal* sigmoid nonlinearities, and contains instead these higher order *synaptic* nonlinearities.

Adding the higher order weights corresponds, in connectionist language, to increasing the complexity of the neural population nodes to become "higher order" or "sigma-pi" units. Clusters of synapses within a population unit can *locally* compute products of the activities on incomming primary connections W_{ij}, during higher order Hebbian learning, to establish a weight W_{ijkl} (see figure 3). These secondary higher order synapses are then used in addition to the synapses W_{ij}, during operation of the overall network, to weight the effect of triple products of inputs in the output summation of the population.

Using only the long range excitatory *connections* W_{ij} available, some number of the higher order synaptic weights W_{ijkl} could be realized locally within a neural population in the axo-dendritic interconnection plexus known as "neuropil" [3]. Only $(N/2)^2$ of these $(N/2)^4$ possible higher order weights are required in principle to approximate the performance of the projection algorithm [6]. The size of our cortical patches is limited by this number, and is itself motivation for modularity.

3.3.1 Hebbian Learning

The minimal network coupling (12) for T results from the projection learning rule (2) when a specific biological form is chosen, in the columns s of P, for the patterns to be stored. Only the higher order weights W_{ijkl} between excitatory populations shown in the biological module (13) are required for approximate pattern storage [6]. This special complex form for P^s and the corresponding asymptotic solutions $X^s(t)$ established are,

$$P^s = \begin{bmatrix} |\vec{x}^s| e^{i\vec{\theta}^s_x} \\ \sqrt{\frac{g}{h}} |\vec{x}^s| e^{i\vec{\theta}^s_y} \end{bmatrix} \quad \Rightarrow \quad X^s(t) = \begin{bmatrix} |\vec{x}^s| e^{i\vec{\theta}^s_x + i\omega^s t} \\ \sqrt{\frac{g}{h}} |\vec{x}^s| e^{i\vec{\theta}^s_y + i\omega^s t} \end{bmatrix} . \tag{15}$$

The phase θ_x, θ_y is constant over the components of each kind of neural population x and y, and differs only between them. This is basically what is observed in the olfactory bulb (primary olfactory cortex) and prepyriform cortex[3]. The phase of inhibitory components θ_y in the bulb lags the phase of the excitatory components θ_x by approximately 90 degrees.

A Hebbian learning rule may be derived from the projection learning rule which allows a network to learn its attractor categories by local self organization of synapses and synaptic clusters according to pre and post synaptic activities experienced during external input forcing. For orthonormal static patterns $\vec{x}^s$, $P^{-1} = P^T$, and the projection rule for the W matrix reduces to an outer product, or "Hebb" rule, and the projection for the higher order weights becomes a *multiple* outer product rule:

$$W_{ij} = \sum_s^{N/2} \alpha^s x_i^s x_j^s \ , \qquad W_{ijkl} = c\delta_{ij}\delta_{kl} - d\sum_{s=1}^{N/2} x_i^s x_j^s x_k^s x_l^s \ . \tag{16}$$

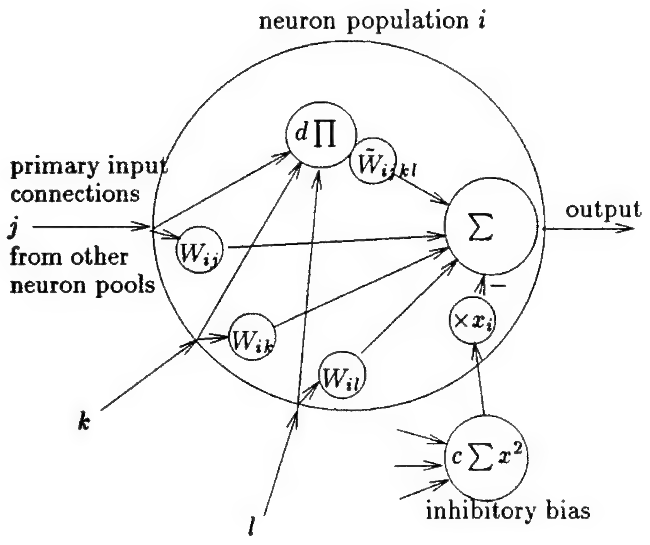

Figure 3: Neural population subnetwork acting as a sigma-pi unit. It uses secondary higher order synaptic weights $\tilde{W}_{ijkl}$ on products of the activities of incoming primary connections W_{ij}, and receives a global inhibitory bias.

When the Hebbian learning rule (16) is used, the higher order weights W_{ijkl} of the network model (13) can be decomposed so that (13) becomes,

$$\dot{x}_i = -\tau x_i - h y_i + \sum_{j=1}^{N/2} W_{ij} x_j + d \sum_{jkl=1}^{N/2} \tilde{W}_{ijkl} x_j x_k x_l - c x_i \sum_{j=1}^{N/2} x_j^2 + b_i \tag{17}$$

where $\tilde{W}_{ijkl}$ comes from the multiple outer product in (16), and $-cx_i \sum_{j=1}^{N/2} x_j^2$ comes from the $c\delta_{ij}\delta_{jk}$ term in (16). This single weight negative term corresponds to a shunting inhibitory bias which depends on the global excitatory activity of the network. "Shunting" here means multiplied by the current average cell voltage x_i of population i. This is an input which is identical for all excitatory neural populations and could be calculated by a single node of the network which receives input from all excitatory populations, as shown in figure 3. Such a node might correspond to one of the nuclei which lie below the prepyriform cortex. These send and receive the diffuse projections required to and from prepyriform cortex. The constants c and d of eq. (17) give the magnitude of the inhibitory bias c and the average higher order weight d. These constants are derived from entries in the normal form matrix A, and, as we will discuss below, $c > d$ guarantees stability of all stored patterns. The greater the bias c relative to d, the greater the "competition" between stored patterns, the more robust is the stability of the present attractor, and vice versa. This is the mechanism employed later in the biological sensory-motor architecture for central control of attractor transitions within modules.

4 Normal Form Associative Memory Modules

As described above, the network modules of the cortical model were developed previously as models of olfactory cortex, or caricatures of "patches"of neocortex [2, 5, 3]. A particular module is formed by a set of neural populations whose interconnections also contain higher order synapses. These synapses determine attractors for that subnetwork independent of other subnetworks. Each module assumes only minimal internal coupling of excitatory and inhibitory neural populations justified by known anatomy in prepyriform cortex, as described earlier.

Figure a.

Low Competition

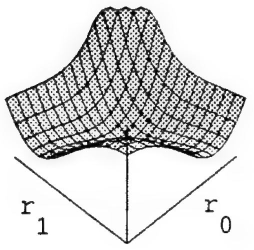

Figure b.

High Competition

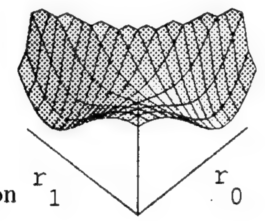

Figure 4: Energy landscape of amplitudes of binary oscillatory unit with no external input. For low levels of competition, there is a broad circular valley. With high levels of competition, there is a deep potential wells on each axis. Phase-locked external inputs simply add a linear tilt to the landscape which will shift a single attractor accross the circular valley at low competition, but can't move it from a potential well at high competition.

In this biological model, the attractors within modules are distributed patterns of activity like those observed experimentally [24]. However, the network is equivalent to the architecture of modules in "normal form" as described above, and may easily be designed, simulated, and theoretically evaluated in these coordinates.

By analyzing the network in the polar version of the normal form coordinates, the amplitude and phase dynamics have a particularly simple interaction. When the input to a module is synchronized with its intrinsic oscillation, the amplitudes of the periodic activity may be considered separately from the phase rotation, and the network of the module may be viewed as a static network with these amplitudes as its activity. We have further shown analytically that the network modules we have constructed have a strong tendency to synchronize as required. An attractor in these winner-take-all normal form cordinates is one oscillator at its maximum amplitude, with the others near zero amplitude.

For the oscillating networks of the present model, we use the normal form for the Hopf bifurcation, which characterizes the genesis of oscillation, and make particular use of it to control bifurcations. A bifurcation is a discontinuous (topologically singular) change in the phase portrait of possibilities for the continuous dynamical behavior of a system (such as the appearance or dissappearance of an attractor) that occurs as a bifurcation parameter reaches a critical value. It is the bifurcation in the vector field of a network module from one to many attractors that effects the essential digitization of the system in time and state, as we will see below.

4.1 A Binary Oscillatory Unit

To illustrate the behavior of individual network modules, we examine a binary (two attractor) module; the behavior of modules with more than two attractors is similar. Such a unit is defined in polar normal form coordinates by the following equations of the Hopf normal form:

$$
\begin{aligned}
\dot{r}_{1i} &= u_i r_{1i} - c r_{1i}^3 + (d - b sin(\omega_{clock} t)) r_{1i} r_{0i}^2 + \sum_j w_{ij}^+ I_j \cos(\theta_j - \theta_{1i}) \\
\dot{r}_{0i} &= u_i r_{0i} - c r_{0i}^3 + (d - b sin(\omega_{clock} t)) r_{0i} r_{1i}^2 + \sum_j w_{ij}^- I_j \cos(\theta_j - \theta_{0i}) \\
\dot{\theta}_{1i} &= \omega_i + \sum_j w_{ij}^+ (I_j / r_{1i}) \sin(\theta_j - \theta_{1i}) \\
\dot{\theta}_{0i} &= \omega_i + \sum_j w_{ij}^- (I_j / r_{0i}) \sin(\theta_j - \theta_{0i})
\end{aligned}
$$

The clocked parameter $bsin(\omega_{clock}t)$ is used to control attractor transitions in the Elman architecture to be discussed later. It has lower frequency (1/10) than the intrinsic frequency of the unit ω_i.

When the oscillators are sychronized with the input, $\theta_j - \theta_{1i} = 0$, and the phase terms $\cos(\theta_j - \theta_{1i}) = \cos(0) = 1$ dissappear. This leaves the amplitude equations $\dot{r}_{1i}$ and $\dot{r}_{0i}$ with static inputs $\sum_j w^{+}_{ij} I_j$ and $\sum_j w^{-}_{ij} I_j$.

Examination of the phase equations shows that a unit has a strong tendency to synchronize with an input of similar frequency. Defining the phase difference, $\phi = \theta_0 - \theta_I = \theta_0 - \omega_I t$ between a unit θ_0 and it's input θ_I we can write a differential equation $\dot{\phi}$ for the phase difference ϕ ,

$$\dot{\phi} = \omega_0 - \omega_I + (r_I/r_0)\sin(-\phi) \ , \quad so \ , \quad \hat{\phi} = -sin^{-1}[(r_0/r_I)(\omega_I - \omega_0)]$$

There is an attractor $\hat{\phi}$ at zero phase difference $\phi = \theta_0 - \theta_I = 0$, and a repellor at 180 degrees in the phase difference equations $\dot{\phi}$ for either side of a unit driven by an input of the same frequency, $\omega_I - \omega_0 = 0$. In simulations, the interconnected network of these units to be described below synchronizes robustly within a few cycles following a perturbation.

If the frequencies of attractors in some modules of the architecture are randomly dispersed by a significant amount, $\omega_I - \omega_0 \neq 0$, phase-lags appear first, then synchronization is lost in those units. An oscillating module therefore acts as a band pass filter for oscillatory inputs.

Thus we have network modules which emulate static network units in their amplitude activity when fully phase-locked to their input. Amplitude information is transmitted between modules, with an oscillatory carrier.

4.2 Attractor Transitions by Bifurcation

For fixed values of the competition, in a completely synchronized system, the internal amplitude dynamics define a gradient dynamical system for a fourth order energy function. Figures 4a and 4b show the energy landscape with no external input for high and low levels of competition respectively. External inputs that are phase-locked to the module's intrinsic oscillation simply add a linear tilt to the landscape.

For low levels of competition, there is a broad circular valley. When tilted by external input, there is a unique equilibrium that is determined by the bias in tilt along one axis over the other. Thinking of r_{1i} as the "acitivity" of the unit, this acitivity becomes a monotonically increasing function of input. The module behaves as an analog connectionist unit whose transfer function can be approximated by a sigmoid. We refer to this as the "analog" mode of operation of the module.

With high levels of competition, the unit will behave as a binary (bistable) digital flip-flop element. There are two deep potential wells, one on each axis. Hence the final steady state of the unit is determined by which basin of attraction contains the initial state of the system in the analog mode of operation before competition is increased by the clock. This state changes little under the influence of external input: a tilt will move the location of the attractor basins only slightly. Hence the module performs a winner-take-all choice on the coordinates of its initial state and maintains that choice independent of external input. This is the "digital" or "quantized" mode of operation of a module. We use this bifurcation in the behavior of the modules to control information flow within the network to be described below.

5 Sensory-Motor Architecture

As a benchmark for the capabilities of the system, and to create a point of contact to standard network architectures, we have shown how a discrete-time "simple recurrent" or "Elman" network architecture [21] can be constructed from recurrently connected oscillatory associative memory modules described by continuous nonlinear ordinary differential equations [14, 11]. The system learns to function as a finite state automaton that recognizes or generates the infinite set of six symbol strings that are defined by a Reber grammar.

The time steps (sensory-motor cycles) of the system are implemented by rhythmic variation (clocking) of the competition bifurcation parameter. This holds input and "context" (sensory) modules clamped at their attractors while 'hidden and output (motor) modules change state, then clamps hidden and output states

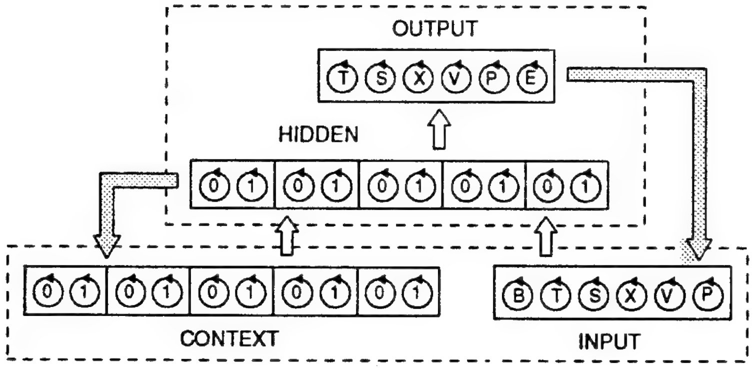

Figure 5: Elman architecture: The input and output modules each consist of a single associative memory module with six oscillatory attractors, one for each of the six symbols in the grammar. The hidden and context modules consist of binary "units" composed of two oscillatory attractors. Dotted lines enclose the two "sensory" and "motor" sets of modules which are allowed to change attractors at alternate peaks of the machine cycle.

while context modules are released to load those states as the new context for the next cycle of input. The dotted lines of figure 5 show these two sets of modules.

We use two types of modules in implementing the Elman network architecture. The input and output layer each consist of a single associative memory module with six oscillatory attractors (six competing oscillatory modes), one for each of the six possible symbols in the grammar. The hidden and context layers consist of binary "units" composed of a two oscillator module with internal competition. We think of one mode within the unit as representing "1" and the other as representing"0" (see figure 5).

The network approximates a static network in its amplitude activity when fully phase-locked. Amplitude information is transmitted between modules, with an oscillatory carrier. If the frequencies of attractors in the architecture are randomly dispersed by a significant amount phase-lags appear, then synchronization is lost and improper transitions begin to occur.

It is the bifurcation in the phase portrait of a module from one to many attractors that contributes the essential digitalization of the system in time and state. The analog mode for a module allows input to prepare its initial state for the binary decision between attractor basins that occurs as competition rises and the double potential well appears.

The feedback between sensory and motor modules is effectively cut when one set is clamped at high competition. The system thus operates in discrete time by alternating sets of transitions between finite sets of attracting states. This kind of alternate clocking and buffering (clamping) of some states while other states relax is essential to the reliable operation of digital architectures, as it is in our modules. The clock input on a flip-flop clamps it's state until its signal inputs have settled and the choice of transition is made with the proper information available. In our simulations, if we clock all modules to transition at once, the programmed transitions lose stability, and we get transitions to unprogrammed fixed points and simple limit cycles for the whole system. This is a strong justification for the use of clamped attractors and clocked cycles.

5.1 Training

During training, the hidden module units are left at zero or negative competition after clamping of input and context. They are thus free to take analog values on a given time step so that a real valued error can be defined and backpropagation may be used to train the system.

If the context units are clamped with high competition, they are essentially "quantized" to take on only their 0 or 1 attractor values, and the feedback connections from the hidden units cannot affect them. While Giles, et. al. generally do not quantize their units until the end of training to extract a finite state automaton, they and others [31] find that quantizing of the context units during training like this increases learning time but produces a network with perfect performance.

We also have the option of leaving the competition within the context units at intermediate levels to allow them to take on analog values in a variable sized neighborhood of the 0 or 1 attractors. Since our system is recurrently connected by an identity map from hidden to context units, it will relax to some equilibrium determined by the impact of the context units and the clamped input on the hidden unit states, and the effect of the feedback from those hidden states on the context states. We can thus further explore the impact on learning of this range of operation between discrete time and space automaton and continuous analog recurrent network.

The ability to operate as an finite automaton with oscillatory/chaotic "states" is thus an important benchmark for this architecture, but only a subset of its capabilities. At low to zero competition, the supra-system reverts to one large continuous dynamical system. We expect that this kind of variation of the operational regime, especially with chaotic attractors inside the modules, though unreliable for habitual behaviors, may nontheless be very useful in other areas such as the search process of reinforcement learning.

5.2 Synchronization, Noise, and Intermodule Communication

An important element of intra-cortical communication in the brain, and between modules in this architecture, is the ability of a module to detect and respond to the proper input signal from a particular module, when inputs from other modules which is irrelevant to the present computation are contributing cross-talk and noise. This is smilar to the problem of coding messages in a computer architecture like the Connection Machine so that they can be picked up from the common communication buss line by the proper receiving module. We are investigating the hypothesis that sychronization control is one way the brain can solve this coding problem.

Because communication between modules in the architecture is by continuous time-varying analog vectors, the process is more one of signal detection and pattern recognition by the modules of their inputs than it is "message passing". This is why the demonstrated performance of the modules in handwritten character recognition is significant, and why we expect there are important possibilities in the architecture for the kinds of chaotic signal processing studied by Chua [22].

We have shown that the dynamic attractors - oscillatory or chaotic - within the modules of this architecture must synchronize to effectively communicate information and produce reliable transitions [10]. In simulations, we have synchronized lorenz and Chua attractors for operation in the architecture using techniques of coupling developed by Chua [22] for secure "broadspectrum" communication by a modulated chaotic carrier wave [11, 13].

We have also shown in these modules a superior stability of oscillatory attractors over static attractors in the presence of noise perturbations with the 1/f spectral character of the noise found experimentally by Freeman in the brain [10]. This may be one reason why the brain uses dynamic attractors. An oscillatory attractor acts like a a bandpass filter and is effectively immune to the many slower macroscopic bias perturbations in the theta-alpha-beta range (3 - 25 Hz) below its 40 -80 Hz passband, and the more microscopic perturbations of single neuron spikes in the 100 - 1000 Hz range. In an environment with this spectrum of perturbation, modules with static attractors cannot operate reliably.

5.3 Attentional Control of Synchrony

The network architecture, shown in figure 6, has been designed so that amplitude codes the information content or activity of a module, whereas phase and frequency are used to "softwire" the network. An oscillatory network module has a passband outside of which it will not synchronize with an oscillatory input. Modules can therefore easily be desynchronized by perturbing their resonant frequencies. They can also be desynchronized by anti-phase inputs as in the models of Koenig, et. al. [?] Furthermore, only synchronized modules communicate by exchanging amplitude information; the activity of non-resonating modules contributes incoherant crosstalk or noise.

The flow of communication between modules can thus be controled by controlling synchrony. By changing the intrinsic frequency of modules in a patterned way, the *effective* connectivity of the network is changed.

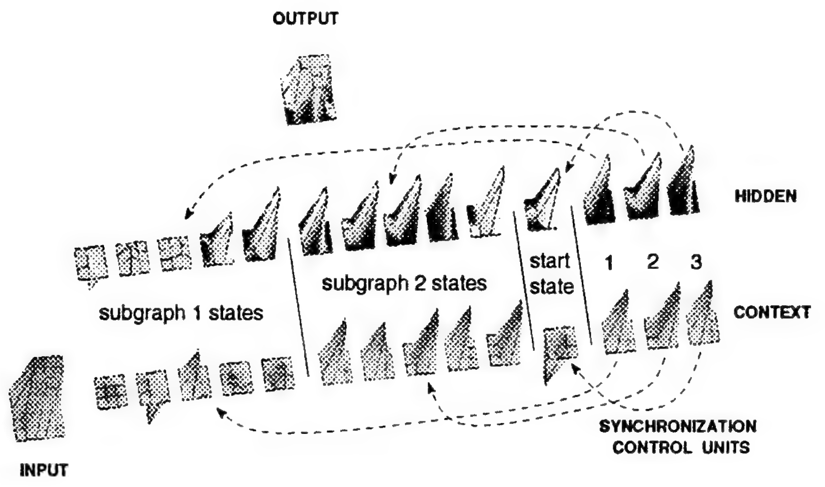

Figure 6: Synchronization control architecture: The input and output modules show the symbol "T" as a distributed attractor pattern. The binary modules of the hidden and context layers show oscillatory attractors in winner-take-all normal form cordinates where one oscillator at its maximum amplitude, with the others near zero amplitude. Activity levels oscillate up and down through the plane of the paper. Dotted lines show control outputs from the synchronization control modules. Control unit two is at the one attractor (right side of the square active) and the hidden units coding for states of subgraph two are in synchrony with the input and output modules. Here in midcycle, all modules are clamped at their attractors.

The same hardware and connection matrix can thus subserve many different computations and patterns of interaction between modules without crosstalk problems.

The crosstalk noise is actually essential to the function of the system. It serves as the noise source for making random choices of output symbols and automaton state transitions in this architecture during reinforcement learning and normal operation after learning. In cortex there is an issue as to what may constitute a source of randomness of sufficient magnitude to perturb the behavior of the large ensemble of neurons involved in neural activity at the cortical network level. It does not seem likely that the well known molecular level of fluctuations which is easily averaged within a single neuron or small group of neurons can do the job. The architecture here models the hypothesis that deterministic chaos in the macroscopic dynamics of a network of neurons, which is the same order of magnitude as the coherant activity, can serve this purpose.

In a set of modules which is desynchronized by perturbing the resonant frequencies of the group, coherance is lost and "random" phase relations result. The character of the model time traces is now irregular as seen in real neural ensemble activity. The behavior of the time traces in different modules of the architecture is similar to the temporary appearance and switching of synchronization between cortical areas as seen in observations of cortical processing during sensory/motor tasks in monkeys and humans [17]. The detailed structure of this apparently chaotic signal and its further use in network learning and operation are currently under investigation.

5.4 Grammatical Inference

We studied the use of these capabilities in the grammatical inference problem by constructing and learning the larger fifteen hidden unit (module) automata studied by Cleermans, et al. This consists of two subgraphs each of which was the automaton learned previously in work described above, and shown in figure 7. Strings of this grammar can contain long embedded sequences of the smaller grammar before the final transition distinguishing which branch you are on appears. These transitions of this grammar were challenging to learn

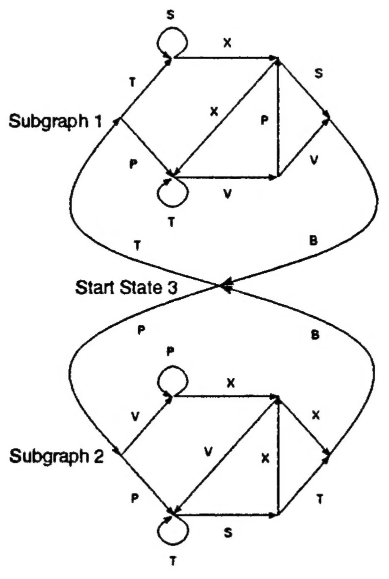

Figure 7: Graph diagram of the automaton emulated by the network to generate the symbol strings of a grammar. It is composed of two subgraphs joined by a start/end state. At each node (network state), one of two symbols (output module attractors) is chosen at random (by crosstalk noise) and fedback as input to the network to direct the next transition of state as shown by the arrows of the diagram.

because of the embedding. Cleermans *et al* had to alter the transition probabilities within the two subgraphs so that the backpropagation algorithm could distinguish the branches during learning.

We solved this learning problem by introducing a control of program flow by selective synchronization [15, 16]. The controler itself is modeled in this architecture as a special set of hidden modules with ouputs that affect the resonant frequencies of the other hidden modules or supply an anti-phase input, as shown in figure 6.

These enforce a segregation of the hidden module code for the subgraph states during training so that different sets of synchronized modules learn to code for each subgraph with the other modules desynchronized by frequency perturbation. The entire automaton is learned with its additional entry and exit hidden module states and with these special hidden modules.

Transitions in the system from states in one subautomaton to the other are made by "attending" to the corresponding set of nodes in the hidden and context layers. This switching of the focus of attention is accomplished by changing the patterns of synchronization within the network.

Varying levels of intramodule competition control the large scale *direction* of information flow *between* layers of the architecture. To direct information flow on a finer scale, the "attention" mechanism selects a subset of modules *within* each layer whose output is effective in driving the behavior of the system.

The system in operation is made to jump from states in one subautomaton to the other by desynchronizing the proper subset of hidden modules. The possibilities for transition of the system are thus be controled by selective synchronization. This control itself is learned by the special hidden units whose output controls the synchrony of these subsets. During training, the control modules learn to respond to the proper input symbol and context to direct the flow of computation to effect the difficult transitions between subgraphs. Viewing the automaton above as a behavioral program, the control of synchrony constitutes a control of the program flow into its subprograms (the subgraphs).

In future work we will investigate the possibilities for self-organization of the patterns of synchrony and spatially segregated coding in the hidden layer during learning. We will explore the use of lateral connections between hidden units to cause competition for synchrony as has been done by Koenig *et al* to see how local spatially segregated coding can be self-organized during learning. Lateral neighborhood connections between hidden units which effect synchrony of neighboring units have been sucessfully implemented in simulations, and certain sections of a large number of hidden units will self-organize into synchrony and take part in the learning of certain subgraphs of the automaton.

6 Computing Resources

Our analytic approach to understanding these networks relies heavily on geometric visualization of network learning and operation in prefered coordinate systems. The computer graphic capabilities of the Silicon Graphics Personal Iris 4D35G workstation purchased by the grant has been invaluable in enabling us to design interactive simulations with graphical display of these geometric representations in order to enhance our intuition and generate new theoretical insights.

We have employed the workstation as a system for simulation and graphic display of network dynamics, where we can vary network parameters (most notably bifurcation parameters) and alter network dynamics in real time. With this capability, we were able to rapidly explore regions of the parameter space, and find where to concentrate our numerical and analytical efforts.

7 Invited Talks and Conferences

1991

Plenary speaker, Applications of Artificial Intelligence and Neural Networks, SPIE, Orlando, Fla., April
Invited lecture, Stanford University, Computer Science Dept., Palo Alto, Ca., April
International Joint Conference on Neural Networks, Seattle, Wa., June
Invited Speaker, Workshop on Complex Dynamics in Neural Networks, Vietri, Italy, June
Analysis and Modeling of Neural Systems 2, U.C. Berkeley, Ca., July
Neural Information Processing Systems - Natural and Synthetic, Denver, Colo., November

1992

Invited speaker, 2nd Int. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan, July
Computation and Neural Systems *92, San Francisco, Ca. July
Neural Information Processing Systems - Natural and Synthetic, Denver, Colo., November

1993

Invited lecture, RICOH Palo Alto Research Center - January
Invited speaker. International Workshop - Self-Organization, Learning and Dynamics in Neural Networks, Toulouse, France, March
Invited lecture, Computer Science Dept., University of New South Wales, Sidney, Austrailia, June
Invited lecture, Computer Science Dept., University of Queensland, Brisbane, Austrailia, June
Invited speaker. Midwest Dynamics Conference, University of California at Berkeley, Berkeley, Ca., October
Neural Information Processing Systems - Natural and Synthetic, Denver, Colo., November

*talk or poster given at all conferences listed

References

[1] B Baird. A bifurcation theory approach to vector field programming for periodic attractors. In *Proc. Int. Joint Conf. on Neural Networks, Wash. D.C.*, pages 1:381–388, June 1989.

[2] B. Baird. Associative memory in a simple model of oscillating cortex. In D. S. Touretsky, editor, *Advances in Neural Information Processing Systems 2*, pages 68–75. Morgan Kaufman, 1990.

[3] B. Baird. Bifurcation and learning in network models of oscillating cortex. In S. Forest, editor, *Emergent Computation*, pages 365–384. North Holland, 1990. also in Physica D, 42.

[4] B. Baird. A learning rule for CAM storage of continuous periodic sequences. In *Proc. Int. Joint Conf. on Neural Networks, San Diego*, pages 3: 493–498, June 1990.

[5] B. Baird. Learning with synaptic nonlinearities in a coupled oscillator model of olfactory cortex. In F. Eeckman, editor, *Analysis and Modeling of Neural Systems*, pages 319–327, 1992.

[6] B. Baird. *Bifurcation Theory Approach to the Analysis and Synthesis of Neural Networks for Engineering and Biological Modeling.* Research Notes in Neural Computing. Springer, 1994. to appear.

[7] B. Baird, W. Freeman, F. Eeckman, and Y. Yao. Applications of chaotic neurodynamics in pattern recognition. In *SPIE Proceedings Vol. 1469*, pages 12–23, 1991.

[8] Bill Baird. Information processing by dynamical interaction of oscillatory modes in coupled cortical networks. In John Taylor, editor, *Complex Dynamics in Neural Networks*, pages 191–208, New York, 1992. Springer.

[9] Bill. Baird and Frank H. Eeckman. CAM storage of analog patterns and continuous sequences with $3n^2$ weights. In D. S. Touretsky, editor, *Advances in Neural Information Processing Systems 3*, pages 192–198. Morgan Kaufman, 1991.

[10] Bill Baird and Frank H. Eeckman. A hierarchical sensory-motor architecture of oscillating cortical area subnetworks. In Frank H. Eeckman, editor, *Analysis and Modeling of Neural Systems II*, pages 96–104, Norwell, Ma, 1992. Kluwer.

[11] Bill Baird and Frank H. Eeckman. A neural network computer architecture for computation with oscillatory and chaotic attractors. In Takeshi Yamakawa, editor, *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan*, pages 981-985, 1992.

[12] Bill Baird and Frank H. Eeckman. A normal form projection algorithm for associative memory. In Mohamad H. Hassoun, editor, *Associative Neural Memories: Theory and Implementation*, New York, NY, 1993. Oxford University Press.

[13] Bill Baird, Morris W. Hirsch, and Frank Eeckman. A neural network associative memory for handwritten character recognition using multiple chua attractors. *IEEE Trans. Circuits and Systems*, pages 667–675, 1993.

[14] Bill Baird, Todd Troyer, and Frank H. Eeckman. Synchronization and gramatical inference in an oscillating elman network. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 236–244. Morgan Kaufman, 1993.

[15] Bill Baird, Todd Troyer, and Frank H. Eeckman. Gramatical inference by attentional control of synchronization in an oscillating elman network. In S.J. Hanson, J.D. Cowan, and C.L. Giles, editors, *Advances in Neural Information Processing Systems 6*, pages 67–75. Morgan Kaufman, 1994.

[16] Bill Baird, Todd Troyer, and Frank H. Eeckman. Gramatical inference with synchronized oscillating associative memories. *Neural Computation*, 1994. in press.

[17] S. Bressler and Nakamura. Inter-area synchronization in macaque neocortex during a visual pattern discrimmination task. In F.H. Eeckman and J. Bower, editors, *Neural Systems: Analysis and Modelling*, page 515, Norwell, MA, 1993. Kluwer Academic Publishers.

[18] Leon O. Chua. Global unfolding of Chua's circuit. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, E76-A(5):704–734, 1993.

[19] Leon O. Chua and Lin Gui-nian. Canonical realization of Chua's circuit family. *IEEE Transactions on Circuits and Systems*, 37(7):885–902, 1990.

[20] Leon O. Chua, Komuro Motomasa, and Takashi Matsumoto. The double scroll family. *IEEE Transactions on Circuits and Systems*, CAS-33(11):1073–1118, 1986.

[21] J.L. Elman. Distributed representations, simple recurrent networks and grammatical structure. *Machine Learning*, 7(2/3):91, 1991.

[22] Tetsuro Endo and Leon O. Chua. Synchronization and chaos in phase-locked loops. *IEEE Transactions on Circuits and Systems*, 38(12):620–626, 1991.

[23] A. K. Engel, P. Konig, C. Gray, and W. Singer. Synchronization of oscillatory responses: A mechanism for stimulus-dependent assembly formation in cat visual cortex. In R. Eckmiller, editor, *Parallel Processing in Neural Systems and Computers*, pages 105–108. Elsevier, 1990.

[24] W.J. Freeman and B. Baird. Relation of olfactory EEG to behavior: Spatial analysis. *Behavioral Neuroscience*, 101:393–408, 1987.

[25] C. M. Gray and W. Singer. Stimulus dependent neuronal oscillations in the cat visual cortex area 17. *Neuroscience [Suppl]*, 22:1301P, 1987.

[26] C.M. Gray, P. Konig, A.K. Engel, and W. Singer. Oscillatory responses in cat visual cortex exhibit intercolumnar synchronization which reflects global stimulus properties. *Nature(London)*, 338:334–337, 1989.

[27] J. Guckenheimer and D. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* Springer, New York, 1983.

[28] Lewis B. Haberly and James M. Bower. Olfactory cortex: model circuit for study of associative memory? *Trends in Neuroscience*, 12(7):258, 1989.

[29] M. W. Hirsch and B. Baird. Computing with dynamic attractors in neural networks. *Biosystems*, 1993. to appear.

[30] R. N. Madan (editor). Chua's circuit: A paradigm for chaos. *Journal of Circuits, Systems, and Computers*, 3(1-2), 1993.

[31] Z. Zeng, R.M. Goodman, and P. Smyth. Learning finite state machines with self-clustering recurrent networks. *Neural Computation*, 1993. to be published.